AFRL-HE-WP-TR-2006-0158

UAV Operator Human Performance Models

Stephen Deutsch

BBN Technologies
10 Moulton Street
Cambridge, MA 02138

September 2006

Final Report for April 2004 to September 2006

Air Force Research Laboratory
Human Effectiveness Directorate
Warfighter Interface Division
Cognitive Systems Branch
WPAFB OH 45433-7604

NOTICE

AFRL-HE-WP-TR-2006-0158 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR

//SIGNED//

DANIEL G. GODDARD
Chief, Warfighter Interface Division
Human Effectiveness Directorate

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
September 2006	Final	April 2004 - September 2006

4. TITLE AND SUBTITLE
UAV Operator Human Performance Models

5a. CONTRACT NUMBER FA8650-04-C-6476

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER 63231F

6. AUTHOR(S)
Stephen Deutsch

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER 28300412

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

BBN Technologies
10 Moulton Street
Cambridge, MA 02138

8. PERFORMING ORGANIZATION REPORT NUMBER

BBN Report No. 8460

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Materiel Command
Air Force Research Laboratory
Human Effectiveness Directorate
Warfighter Interface Division
Cognitive Systems Branch
Wright-Patterson AFB OH 45433-7604

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/HECS

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AFRL-HE-WP-TR-2006-0158

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

AFRL/PA Cleared on 2 February 2007, AFRL/WS-07-0207.

14. ABSTRACT
The Distributed Operator Model Architecture (D-OMAR) was used as the software environment in which to build an Unmanned Aerial Vehicle (UAV) test bed. Using D-OMAR, models were developed for UAVs each with a sensor package that included daytime-TV and infra-red cameras. Human performance models were developed for the Aerial Vehicle Operator (AVO) and the Sensor Operator (SO). The modeled workplace from which the AVO and SO managed a mission included positions for the AVO's control of the vehicle the SO's use of the sensor package. With the basic elements of the test bed defined, a use case was developed based on a scenario (Petkosek, Warfield, and Carretta, 2005) at a commercial airport. The AVO model maintained the UAV flight path and the SO, reading from a list of Essential Elements of Information (EEI), conducted and interpreted the sensor observations of the modeled airport scene coordinating them with a Multi-Function Operator (MFO), the third member of the UAV mission team. The operations of the AVO and MFO are described and additional detail is provided on the SO model that describes multitasking, the modeling of individual difference, the modeling of episodic and declarative memory, and model robustness. The report concludes with suggestions that the UAV test bed might be employed to examine workplace design and operating procedures with the goal of improving UAV mission performance, and reducing staffing and required training time.

15. SUBJECT TERMS Unmanned Aerial Vehicle (UAV), Aerial Vehicle Operator (AVO), Sensor Operator (SO), Multi-Function Operator (MFO), Human Performance Modeling, Situational Awareness

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Micheal J. Blake
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR	26	**19b. TELEPHONE NUMBER** *(include area code)*

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. 239.18

THIS PAGE LEFT INTENTIONALLY BLANK

Table of Contents

Table of Figures

THIS PAGE LEFT INTENTIONALLY BLANK

1 Introduction

Unmanned aerial vehicles (UAV) are being employed in increasing numbers in military airspaces and are anticipated to begin to find a role in commercial airspaces as well. From a controls perspective, today's UAVs are idiosyncratic with each UAV type presenting a unique set of operating instruments and procedures to their operators. Accident rates are high when compared to piloted military and commercial aircraft (Manning, Rash, LeDuc, Noback, & McKeon, 2004; Williams, 2004), but while the accidents result in high operating costs, fortunately they do not involve the loss of life. These observations suggest that there is much room for improvement in the design of UAVs, the workplaces from which UAVs are operated, and in the procedures employed by their operators (c.f., McCarley & Wickens, 2005). These improvements can reasonably be expected to lead to the more effective and less costly use of UAVs by the military and are essential if UAVs are to be employed in commercial airspaces. Improved UAV mission effectiveness, and reduced staffing and training time will translate into significant savings.

There are several fronts on which action is needed to bring about the potential improvements in UAV effectiveness at reduced costs. Our focus in the current research project has been comparatively narrow and yet it has the potential to play an important role in UAV systems and procedure development and UAV operations. In particular, we were tasked to build human performance models for UAV operators and construct scenario elements for a particular UAV mission. The models have the potential to play a role in many aspects of UAV workplace design and operating procedure development. They can play a role in the development and evaluation of new designs for UAV operator workstations; they can play a role in the development and evaluation of new operating procedures; they can play supporting roles in UAV training environments; and they can be used to help better understand and mitigate the sources of human error leading to incidents and accidents.

In the present task we developed a UAV-model test bed, an initial set of UAV operator models, and a scenario that served as a use case to facilitate model development. Our focus was on the model for the sensor operator (SO), but models for the aerial vehicle operator (AVO) and multi-function operator (MFO) were developed as well. We begin, in Section 2, by describing the simulation framework that was employed to provide the *UAV test bed* having the potential to examine new approaches to UAV operations using human operator models for the UAV aircrew. In Section 3 we outline the scenario that was selected as the *use case* to support model development. The section also describes the human performance models and the models for the entities that made up the scene observed by the SO in the use case scenario. Section 4 describes the development of the models for the UAV aircrew—the AVO, the SO, and the MFO. The development of the SO model led to important findings with respect to improved model performance and more specifically model robustness—those related to modeling individual difference, episodic memory, and model robustness are discussed in some detail. Model robustness, was previously addressed in more detail in the project's first year report (Deutsch, 2005a). This report concludes with Section 5 that outlines of the applications for which the UAV test bed might be used in the future.

2 The UAV Test Bed

The planned test bed in which the UAV operator models were to operate was a simulation environment that included multiple UAV simulators. The Multiple Unified Simulation Environment (MUSE) capable of simulating most currently active UAV types was to be used as the UAV simulator for the test bed. While a single copy of the MUSE was to be used initially, multiple copies could then be used to form a larger simulation environment. Within the MUSE environment, a Control Station Surrogate (CSS) provides a workplace at which UAV operators can control a UAV in executing a simulated mission—it is a real-time simulation environment that provides human-in-the-loop simulation. Our project goal was to develop UAV operator models and thereby extend the MUSE simulation environment to also be capable of model-in-the-loop simulation. Within the MUSE environment, the goal was to complement the CSS operating as a human operator workplace for the MUSE with the newly developed UAV human operator models that directly control a UAV in the MUSE simulation environment much as the human operators do using the CSS.

Unfortunately, gaining access to a MUSE simulator to support model development in a timely and cost-effective manner became a problem that could not readily be resolved. To address this problem as it surfaced, we elected early on to develop a laptop computer test bed to support initial UAV operator model development. Electing the laptop test bed approach allowed us to readily move ahead on achieving project goals related to developing the UAV operator models. Over the term of the research effort, the decision to go forward with the use of the laptop-based UAV test bed avoided the expense of the acquisition of a MUSE simulator and the cost of the development of the interface between the UAV operator models and the MUSE simulator. UAV operator model development was accomplished without requiring access to the MUSE.

The D-OMAR[1] simulator (Deutsch, Adams, Abrett, Cramer, & Feehrer, 1993; Deutsch, Hudlicka, Adams, & Feehrer, 1993; Deutsch & Adams, 1995) that was used for UAV operator model development then served as the interim UAV model test bed. In using the D-OMAR simulator, we were able to take advantage of the ability of the simulator to run in fast-time. This had the important advantage of saving considerable time during model development due to the significant reduction in run times for the basic use case scenario trials. The use case scenario, covering a little over eighteen minutes in real time, completed in just over twenty-one seconds running in the fast-time simulator.

The D-OMAR simulator operating as the UAV test bed provided the framework for the development of the human performance model and the active entities (e.g., UAVs, aircraft, fuel truck, etc.) in the use case scenario. Early in the development cycle we ran two UAV models; for the use case scenario that was selected for further developed, only a single UAV model was required. Should a MUSE simulator become available at some future point in time, there is no reason why the present D-OMAR-based UAV operator models could not readily be adapted to interface to a MUSE UAV model. To facilitate running with the MUSE simulator, the same D-OMAR models (adapted to interface to

[1] Source code and documentation for D-OMAR is available at omar.bbn.com.

the MUSE) can be used with the D-OMAR simulator then running in real-time to accommodate real-time MUSE operation.

3 The UAV Scenario

With the test bed in place, the next important decision was that of the use case scenario to be used to drive UAV operator model development. In the scenario selected for the use case, the UAV team conducted a surveillance operation at a commercial airport where several armed agents were loading and fueling an aircraft (referred to hereafter as the *target aircraft*) in preparation for their departure. The scenario was derived from related UAV crew modeling research effort by Petkosek, Warfield, and Carretta (2005). Figure 1 reproduced from Petkosek et al. (2005) provides a view of the scene at the airport.

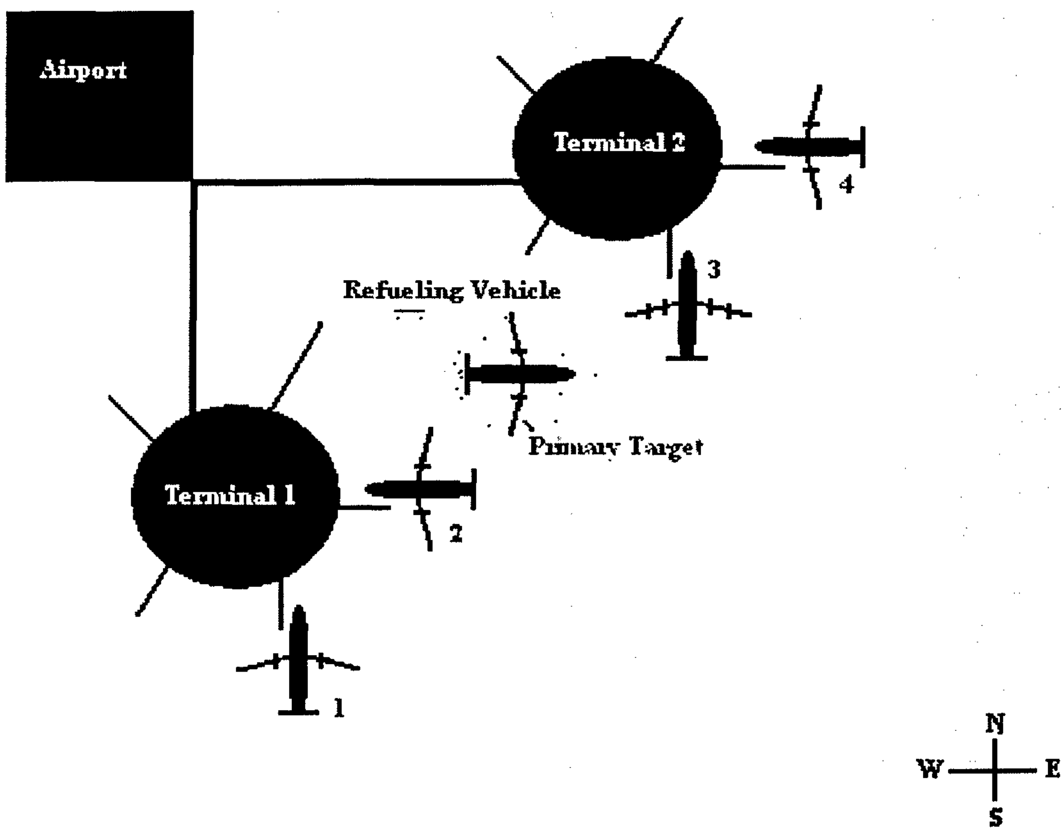

Figure 1 The target aircraft at a commercial airport (Petkosek et al., 2005)

The implementation of the scenario required models for the various human players and active vehicles some of which are seen in Figure 1. The human players included the armed agents controlling operations related to the target aircraft, the personnel fueling and loading or unloading the aircraft, the aircrews for the target aircraft as well as the other aircraft at the adjacent terminals, and the air traffic controllers managing local aircraft operations including that of the departing target aircraft. The scenario vehicles included the aircraft positioned at the terminals and the fuel and cargo trucks servicing the aircraft.

The basic outline for the activities of the scenario was relatively straightforward. A group of armed agent had control of the target aircraft at a commercial airport. The target aircraft was being refueled and there were cargo trucks being used to either to load materials into the aircraft or to obtain supplies from the aircraft. When these operations were completed the aircraft taxied to a runway and then departed the airport. Subsequent sections on the development of the models will provide further details on the actions of the various players in the scenario.

The decision to pursue the laptop version of the UAV test bed allowed us to shed some tasks, but also required additional tasks be taken on. The main task shed, for the present, was the construction (or adaptation) and use of an interface to the MUSE. On the other hand, it was necessary to provide a model for the UAV and the UAV workplace, and models of the scene that would be the subject of the surveillance by the UAV operators as the use case scenario played out. Fortunately, we were able to draw on the work of a previous study in which we examined the error sequence leading to an accident at the Charlotte/Douglas International Airport on July 2nd, 1994 (Deutsch, 2005b; Deutsch & Pew, 2004a). The airport model used for the use case scenario was developed as an extension to the airport model for the previous research effort. The minor extensions to the airport model included the modeling of the terminals for the parked aircraft, the addition of the taxiways servicing the terminal area and Runway 5, and the addition of Runway 5 used by the target aircraft for its departure. The previous research effort had focused on an approach to Runway 18R.

3.1 The UAV and UAV Workplace Models

In the absence of the MUSE we had to develop a UAV model; in the absence of a CSS we had to develop a workplace model, but a workplace model for human performance models rather than the human operators supported by the CSS. Our starting point for the UAV model was our basic commercial aircraft model. We simply adjusted the aircraft model performance parameters to conform to the basic operating envelop of a UAV.

The UAV workplace model was derived from our commercial aircraft flight deck model. The captain's flight deck position became the AVO's workplace; the first officer's flight deck position became the SO's workplace. With the focus of our human performance modeling effort on the SO, we were less concerned with the fidelity of AVO's workplace. In particular, there was a Flight Management Computer (FMC) and a Mode Control Panel (MCP) present in the commercial flight deck model that we simply carried over to the UAV workplace model. With the flight path programmed in the FMC-MCP, the AVO simply has to monitor the UAV's progress along the flight path as displayed on a horizontal situation indicator (HSI). Had any course corrections been required, they could have been established by the AVO using the FMC-MCP as is currently done by the pilot models in the commercial aviation scenarios.

Developing the model for the SO workplace required more work. The workplace modeling effort involved providing the SO model with workplace controls to select and operate the sensors. A daytime-TV camera and an infra red (IR) camera were added to the vehicle model. A selector was provided to enable the SO model to choose the camera to be active. Initial camera positioning was set by the SO by entering a latitude and longitude; zoom was controlled by a lever. It was sufficient to lock the sensors on the

selected lat-long position for scenarios explored. Lastly, it was necessary to provide the SO (as well as the AVO and MFO) with the ability to view the sensor screen and see the objects in the field of view for the particular selected sensor. Visual capabilities were established such that the models were able to take in the visual scene as well as direct their gaze toward particular simulation objects in the field of view.

3.2 Models for the Scenario's Observed Storyline

Previous work provided a model for the Charlotte/Douglas International Airport that included runways and taxiways to which we added two terminals to create an airport environment similar to that as outlined in Petkosek et al. (2005). Figure 2 provides a D-OMAR screen view of the airport's terminals, runways, and taxiways. It was then an easy matter to include the four commercial aircraft and the target aircraft with their respective aircrews as also seen in Figure 2. The screen view is from late in the scenario showing the target aircraft on Runway 5 about to initiate its takeoff roll. While the earlier Charlotte based research effort provided aircrews, aircraft, and air traffic controllers, it was an approach and landing scenario, hence it was necessary to add taxi and take-off procedures for the aircrew and controllers to support the departure of the target aircraft.

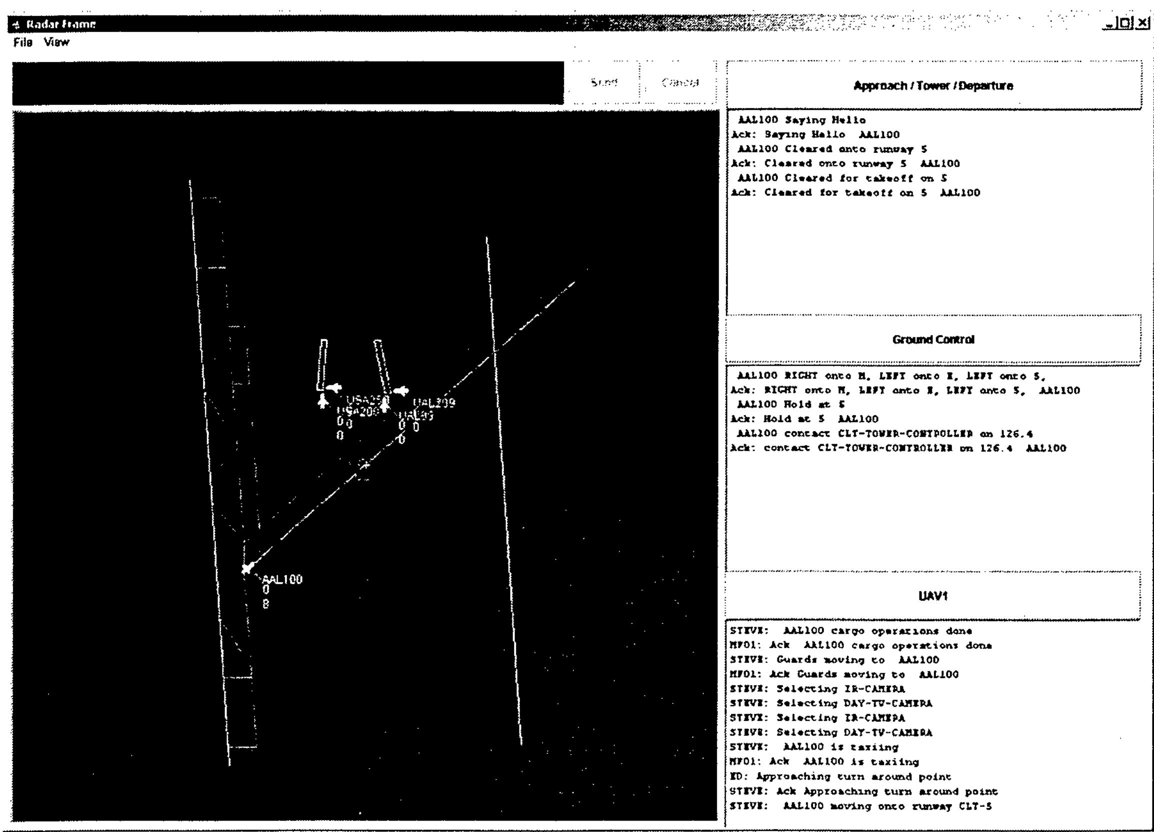

Figure 2 D-OMAR screen shot near the end of the Petkosek et al. (2005) scenario

With respect to our use case scenario, the UAV operators were concerned with monitoring the movements of the aircraft. The aircrews and air traffic controllers, while necessary to support the aircraft's movements, were not direct observables for the UAV

operators. On the other hand, there were a number of scenario players that had a central role in the observations made by the UAV operators. These included the armed guards and the fuel and cargo truck operators all conducting operations related to the target aircraft.

D-OMAR models were constructed so that a fuel truck and one or more cargo trucks could take part in the scenario. An operator model for the fuel truck performed the basic functions of fueling the aircraft and driving the truck as necessary between aircraft. In like manner, operators were provided for the cargo trucks. The Petkosek et al. (2005) scenario had a single cargo truck from which the aircraft was loaded. We added the flexibility to have multiple trucks with operators that might be either loading or unloading the aircraft. The purpose was to provide a more varied scene to be observed and interpreted by the UAV operators.

The last group of models for the UAV operators to observe was the contingent of armed agents. Provision was made to allow an arbitrary number of agents to take part in the scenario. The contingent of agents had a leader who was responsible for orchestrating the operations related to the target aircraft.

3.3 The Storyline Observed by the UAV Operators

As the storyline opened, armed guards had secured the target aircraft at a commercial airport—the Charlotte/Douglas International Airport model. There were four additional commercial aircraft parked at the gates of adjacent terminals. A fuel truck was present and an operator was about to begin the process of refueling the target aircraft. Depending on the particular scenario, there were one or more cargo trucks with additional operators about to begin the processes of moving supplies either from the trucks to the aircraft or possibly from the aircraft to the trucks. A truck to which aircraft supplies had been off-loaded would presumably be of continuing interest to the UAV operators, in contrast to truck that was abandoned having been off-loaded.

As the scenario progressed, the operators fueling and loading or offloading of the aircraft initiated and subsequently completed their operations. The leader of the armed guards (referred to as the *leader* hereafter) monitored these activities and upon completion, directed the guards to enter the target aircraft in preparation for its departure. The leader then notified the aircraft's captain that they were ready to depart. At this point, the captain conferred with the Charlotte ground controller and followed standard commercial aircraft operating procedures in taxiing to Runway 5, conferred with the tower controller and when cleared, proceeded with their take-off roll.

Figure 2, providing a D-OMAR screen view of the airport area, was captured at a point very close to the end of the Petkosek et al. (2005) scenario. At the moment of the screen view, the target aircraft was just starting its takeoff roll on Runway 5. The four remaining commercial aircraft can be seen parked at their respective terminals. The panels on the right provide a trace of the communications among the several groups taking part in the scenario over several minutes leading up to the current time. The lower panel on the right labeled UAV1 records the conversation of the UAV crew. In the dialogue, Ed is the AVO, Steve is the SO, and MFO1 is the MFO. Following the UAV crew dialogue, the SO has announced the observation of the completion of the loading process for the target

aircraft. The SO then cycles between the IR and TV sensors checking for the startup of the target aircraft engines and monitoring activities surrounding the target aircraft.

As events further unfold, the SO does not catch the startup of the engines using the IR sensor, but does see that the guards are moving to board the aircraft and subsequently notes that the aircraft has started its taxi maneuvers. Hence, the SO abandons the use of the IR sensor to focus on monitoring the further actions of the target aircraft using the TV sensor to monitor and report on the aircraft's movement toward and onto Runway 5. In each case, as the SO reports on his or her observations, the MFO acknowledges the communication and the AVO attends the communications as well.

The Ground Control and Approach/Tower/Departure panels trace the communications between the target aircraft's aircrew and the succession of ground, tower, and departure controllers as the aircraft departs from the airport. The aircrew and controllers follow standard operating procedures in managing the departure of the target aircraft.

4 Modeling UAV Team Operations

Within the UAV test bed, individual UAV operations were each controlled from a modeled two-person workplace with workstations for an AVO and an SO. The AVO executed the fairly simple tasks of monitoring the flight of the UAV along a preprogrammed route and communicating with the other UAV team members, while most of the work of completing the mission fell to the SO in conducting the observations of the activities surrounding the target aircraft at the airport. In addition, there was an MFO who supported operations at the workplace for the single UAV operating in the current scenario. The MFO would typically support operations at multiple workstations and in a more extensive scenario might have moved from one UAV workplace to another as the situation demanded. In this section, we will look briefly at the modeling of the behaviors of the AVO and the MFO, but outline in greater detail the innovative aspects of the modeling of the work of the SO.

4.1 The Aerial Vehicle Operator Model

The AVO was responsible for piloting the UAV along a prescribed course suitable for conducting the necessary observations. Gluck, Ball, Krusmark, Rodgers, and Purtee (2003) have developed a human performance model for an AVO as the pilot for a simulated Predator. Unlike the Predator which must be manually piloted, our UAV model, derived from a commercial aircraft model, was equipped with a Flight Management Computer and a Mode Control Pane. We took advantage of the capabilities of the FMC-MCP equipped UAV model to pre-program the required mission route. Figure 3, adapted from Petkosek et al. (2005), portrays the route for the surveillance mission that the UAV that was programmed into the FMC-MCP. The demands of the scenario were such that the AVO did not have to intervene to adjust the route. With the mission route preprogrammed, the AVO simply had to monitor the progress of the UAV along the prescribed route as it was portrayed on the horizontal situation indicator (HSI). As the UAV progressed along the route, the AVO made call-outs of the point-of-closest-approach and the turn-around points based on observations of the HSI's plan view display at the AVO workplace. One of the callouts can be seen in the trace for the UAV crew dialogue in the lower right hand panel of Figure 2. For scenarios or those portions

of scenarios for which active AVO control of the UAV is not required, the FMC-MCP combination stands in to significantly reduce the workload for the AVO.

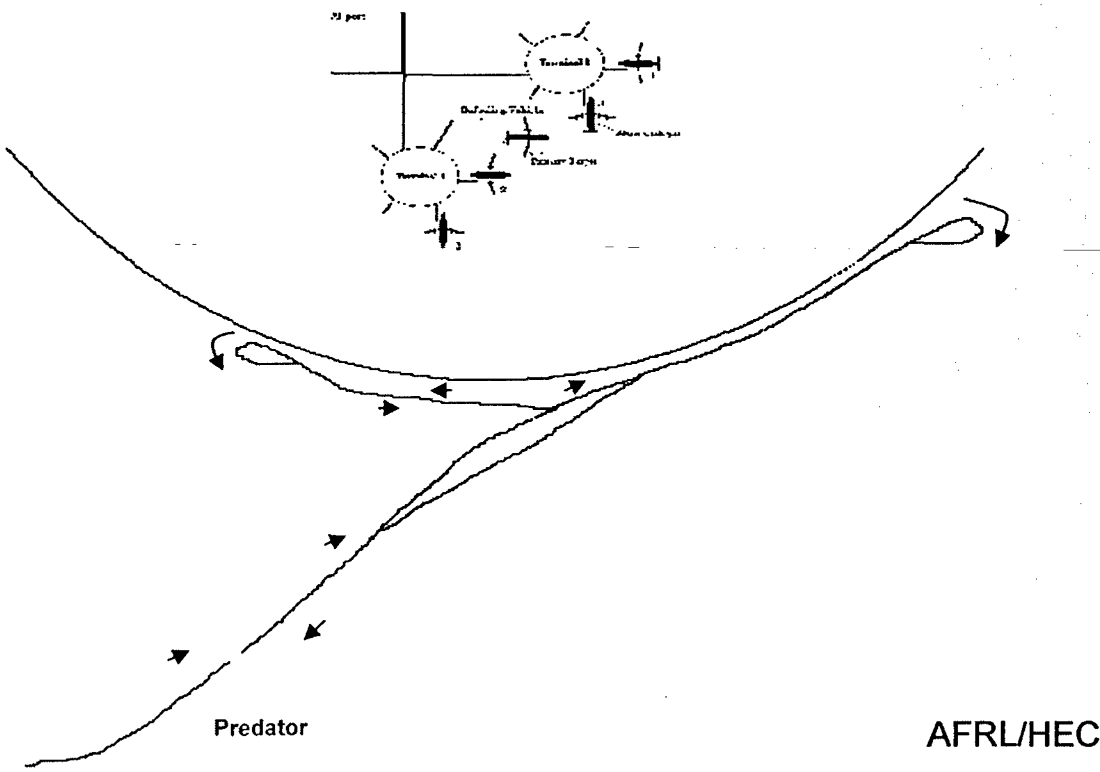

Figure 3 UAV Mission Plan (Petkosek et al., 2005)

4.2 The Sensor Operator Model

The tasking for the UAV mission was governed by a modeled text document containing the Essential Elements of Information (EEI) that defined mission objectives and provided available information essential to support EEI processing. The SO managed the observations using TV and infra-red (IR) sensors that were slaved to move together enabling the TV sensor to be used to target the IR sensor for IR observations. Using information that was read from the EEI document, the SO established the initial pointing of the sensor package by entering a latitude and longitude for the airport and adjusting the senor zoom as required.

In the Petkosek et al. (2005) scenario as developed, there were six EEIs: (1) identify the target aircraft among the aircraft at the airport; (2) count the armed agents; (3) monitor the fueling of the aircraft; (4) monitor the loading and or unloading of the aircraft; (5) check the target aircraft's engines for start-up; and finally (6) conduct surveillance. The SO read and interpreted the requirements of the EEIs and conducted the necessary sensor operations to complete the mission. Each of the EEIs was accomplished using the TV sensor except the checking of the target aircraft's engines for start-up that required the use of the IR sensor. As the SO progressed through the processing of the EEIs, the SO communicated his or her findings to the MFO and AVO.

4.2.1 Characteristics of the Essential Elements of Information

We can begin to describe the SO's execution of the EEIs by looking at the characteristics of the individual EEIs that impacted their execution. The individual EEIs ranged from notably simple (e.g., count the armed agents) to potentially quite complex—the surveillance of the airport could well play out in any number of ways. There were a number of players in different roles present at airport, each with several options on what they could do and there was the possibility for new players to arrive on the scene.

In addition to the complexity dimension, there is an important temporal dimension. Some of the EEIs were completed immediately (e.g., the identification of the target aircraft, and once again, the counting of the armed agents), while most of the EEIs involved the monitoring of events that had an indeterminate timeframe requiring that they be attended over an extended time period. Hence, the SO was frequently multitasking—processing more than one EEI at a time.

Finally, there were instances in which there were dependencies among the EEIs. The target aircraft had to be identified before any of the other EEIs could be pursued. Events detected in executing one EEI could also impact the pursuit of another EEI. In the case that the aircraft was observed moving toward a runway, had the engine start-up not been detected using the IR sensor; it was clearly no longer necessary to pursue that EEI. With the departure of the aircraft from the airport, the surveillance EEI might well be completed unless one of the trucks was loaded with materials off-loaded from the aircraft. The potential for complexity in the surveillance EEI was quite open-ended.

Incompatibilities among the EEIs were a counterpoint to dependencies. While the TV sensor was used across most of the EEIs, detection of engine start-up required the use of the IR sensor, in effect, isolating the monitoring of engine-startup from all the other observations, several of which proceeded concurrently.

4.2.2 Sensor Operator Tasks and Goals

In building the SO model, we posited an explicit association with tasks that mapped to particular EEIs. For most of the EEIs, there was an alignment between a task and the set of operations demanded of the SO as he or she worked through an individual EEI—each a well-defined, notably compact unit of work, allowing that some were not immediately completed. (The one exception among the six EEIs outlined above was the surveillance EEI that was necessarily decomposed into several concurrent tasks in the model.) Hence, in terms of the SO's work, the processing of an EEI constituted a task guided by the goal of completing the particular objective dictated by the EEI. In the model, the goal associated with the EEI was represented as a Simulation Core (SCORE) language goal with a SCORE language plan that included sub-goals as necessary. An SO model's task consisted of the work of the goal's procedures that governed the actions to complete the processing of an EEI.

The reading of the EEIs by the SO to establish the tasks associated with the processing of each EEI constituted a separate task with its own SCORE language goal and procedures. Via this task, the text for an EEI was read and the procedures for processing the EEI were then launched. Having read an EEI and launched the procedures to accomplish it, the SO was then ready to read the next EEI. Just how the task of reading the EEIs and the tasks

of executing the multiple EEIs played out is discussed in more detail in the next section on the modeling of multitasking in the work of the SO..

4.2.3 Multi-tasking by the Sensor Operator

As we thought about modeling multitasking, we were concerned with the SO's work in pursuing the execution of a UAV mission's multiple EEIs. An SO's thought processes in shifting attention from one EEI to another EEI may sometimes be conscious, thoughtful decision-making that can be modeled as just that—explicit decision-making task steps. On the other hand, most SOs are skilled operators for whom much of their action selection is automatized (Logan, 1988a; 1988b; Bargh & Chartrand, 1999). Our primary concern in the modeling effort was the emulation of the fluid, automatized interleaving of the work of multiple EEIs being processed concurrently—the work of the skilled SO—rather than the explicit, thoughtful decision-making required of the less skillful operator.

The work of an EEI included the reading of the textual material defining the work to be accomplished, the mapping the work defined by the EEI to the operations to be performed, and the execution of the required operations. Given that a UAV mission virtually always includes multiple EEIs, we can broadly define bounding approaches to accomplishing the necessary work. A first approach, what we termed the read-process approach, can be defined as read-process in the sense that each EEI is read and executed in turn. At the other extreme is the read-read approach, where an SO might read through all of the EEIs and then proceed with their execution—the EEIs are all read up front and then processed with much resultant concurrency.

In general, the read-process approach will break down simply because the SO will encounter EEIs that can not be immediately resolved and hence, would prevent starting the processing of subsequent EEIs—processing essential to further information gathering. It was thus necessary to read ahead and this of course led to the concurrent execution of multiple tasks. Self evident in its shortcomings, the read-process was not explored using the model. At the other extreme, the read-read approach was explored in the modeling, followed by an examination of the trace of the behaviors produced that showed anomalies in task execution. The exploration of the aspects of the model that drive the middle ground in behaviors between read-read and read-process is discussed in the next section.

4.2.4 Conflict Resolution in Sensor Operator Task Execution

Goals and procedures, the procedural language constructs that drive a model's task execution, are each defined as concepts in the Simple Frame Language (SFL), a direct descendent of KL-ONE (Brachman & Schmolze, 1985). As such, the goal and procedure objects (hereafter, we will simply use procedures to refer to both goals and procedures) reside in a multiple inheritance hierarchy much like the objects in an object-oriented programming language. In the SCORE procedural language, conflicts can be established among procedures. By using the procedures' inheritance relationships, conflicts can be set up between classes of procedures. As an example, the read-process approach to EEI processing could readily have been established using a conflict between the reading of an EEI and a concept subsuming the procedures for completely processing each EEI. An EEI would be read and the processing of the EEI initiated. The established conflict would prevent the reading of the next EEI until processing of the first EEI was completed. As

noted above, this would not work well—executing an EEI that is not immediately completed prevents further EEI processing—and was not pursued in the modeling effort.

What this implies is that the structure of the procedure hierarchy with respect to conflicts between procedures drives the fine structure in the order of task processing in important ways. Changes in the conflict structure for procedures lead to changing patterns of task execution. A coarser concept hierarchy for conflicts yields a more rigid and more orderly execution of a task by inhibiting the interruptions of one task by another. Through experience, the conflict structure might, in fact, become more finely grained in its restrictions thereby enabling the more complexly structured interleaving of competing tasks that fosters improved performance.

Before looking at the impact of refining the conflict structure we briefly review the role played by the baseline conflict structure as we began this particular line of investigation. In the past, the conflict structure has focused on *resources* and *protocols*. Resources are most often perceptual or motor processing centers. A task requiring the fine adjustment of a dial to establish a precise value will require the vision system and usually the dominant hand. The task can only go forward if both resources are available or the task has sufficient priority to commandeer the use of the required resources if presently employed by another task. The deliberate act of setting the dial would typically have higher priority than say, a background instrument scanning task. These conflicts are resolved near the leaves of the procedure hierarchy where the low-level resources are employed. The task with higher priority gains access to the necessary perceptual and motor resources and its execution proceeds.

Protocol conflicts guide higher-level behaviors and are resolved at a higher level in the procedure hierarchy—closer to the goals for a task. For example, on a commercial aircraft flight deck, the aircrew will defer their in-process intra-crew conversation to attend to an air traffic controller communication. The conflict is defined and resolved near the root of the task hierarchy where the conduct-ATC-communication procedure is established with a higher priority than the conduct-intra-crew-conversation procedure.

The implementation of the conflict resolution strategy in a model is quite simple. When a new procedure starts up, it immediately determines if there is a running procedure with which it potentially conflicts. If not, it simply proceeds with its execution. If there is a conflict, the priority of the new procedure is compared with that of the conflicting procedure. If the new procedure's priority is higher, the new procedure continues execution and the running procedure is suspended. If its new procedure's priority is lower, the new procedure may be either suspended or terminated. When a running procedure terminates and there was a procedure that it caused to be suspended, that procedure will resume execution.

4.2.5 Findings Related to Individual Difference and Learning

In our earlier modeling work, the conflicts defined in the procedure hierarchy focused on regions near the root of the goal-procedure hierarchy to establish operational protocols and at the leaves of the hierarchy to govern access to perceptual and motor resources. In the current research effort, our attention was drawn to the, until now, neglected middle ground in the hierarchy. Starting from the baseline conflict structure, we knew that we needed to protect EEI reading vis-à-vis the initial launch of the required EEI processing

so that the initial read of an EEI could be completed. Yet, examination of the traces of the trials with this quite relaxed structure uncovered some questionable model behaviors. A model would read a latitude (for pointing the sensor package) from an EEI, dial in the latitude at the console and then jump off to an unconnected step in the processing of another EEI, maybe even reading a new EEI, before returning to set up the longitude associated with the sensor pointing operation.

The behaviors were not wrong; they were just not the likely behaviors of a good SO. The setting of the latitude *and* the longitude was a task that needed to be protected as a single integrated operation. This was readily accomplished by minor adjustments in the conflict structure for the involved procedures. Adding a conflict between the initial processing of each EEI and the reading of the next EEI secured the initial processing of an EEI, for example, the setting of the latitude *and* the longitude, as a unitary operation. The procedure hierarchy made it possible to establish the conflict once for an initial EEI processing procedure with each particular initial EEI processing procedure inherited as a parent. With the new conflicts in place, the SO would read an EEI and complete a first pass at processing the EEI before continuing either by reading another EEI or processing another open EEI.

The conflict structure and the supporting conflict resolution strategy had always been an essential element in the models for resolving resource conflicts and establishing operator protocols, and as such, central to modeling human-like multi-task performance. As we pursued the process of examining the structure of the middle level conflicts in the procedure hierarchy and adding, removing, or adjusting the pattern of conflicts, we uncovered a broad range in the manner in which the component procedures for multiple tasks could be interleaved to successfully complete the required set of tasks associated with processing the EEIs. What we have now found is that variation in the conflict structure can lead to alternate paths of execution—variety in the interleaving of processes to complete multiple ongoing tasks.

What is new is the conflict structure's potential role in establishing individual differences in task execution as well as possibly playing a role in an individual's learning over time where what is learned is how to more finely structure the execution of multiple procedures, as expressed in the conflict structure, to more effectively complete multiple ongoing tasks. The findings suggest the particular structure of the conflicts between procedures is a contributor to the realization individual differences in human performance. Moreover, there is the suggestion that changes in the conflict structure over time might be one aspect of an individual's progress in learning to more readily and robustly achieve the successful completion of multiple ongoing tasks. A subject might start the learning of a new set of tasks with a highly restrictive conflict structure and through refinement of the structure—the selective additional, removal, or adjustment of conflict elements—evolve a more sophisticated processing of multiple ongoing tasks.

The necessary issue to address at this point is a difficult one: Is the approach to conflicts resolution exhibited in the model relevant to how people might interleave the tasks that they perform in addressing multiple tasks? Do the mechanisms for conflict resolution in the model have anything to say about the mechanisms that people rely on as they address conflicting task demands? An evolutionary argument is perhaps the strongest one in its favor. The conflict resolution strategy as developed in the model is essentially a skill-

based behavior that does not require conscious, thoughtful processing. It is readily possible to imagine a very primitive organism with a single resource and two conflicting demands on that resource. Associating an inherent or learned priority with the demands is a small innovation that might readily emerge to help our ancient little creature to make the necessary choices to survive.

In comparison with the thorny issues of tasks as the objects of a reasoning process, perhaps through rules operating on these representations (Clark, 1997) of the tasks, to allocate resources among tasks, the conflict resolution strategy is simplicity itself. And yet, the markedly simple conflict resolution strategy, as well as addressing resource capture, deals effectively with establishing complex high-level protocols, and is now suggested as perhaps contributing to individual difference and learning in task execution. Once the simple conflict resolution strategy is in place, it is not hard to imagine an exaptation path leading to, in this case, more complex applications. By way of contrast, it is difficult to conceive of how a rule-based executive might have emerged so early on in the evolutionary process given the late start its propositional foundation necessarily implies.

4.2.6 Episodic and Declarative Memory in the Sensor Operator Model

Virtually all of the system diagrams for human performance models include a box for working memory. The working memory box typically is the home for at least, and sometimes exclusively, short-term declarative memory. In some cases, forgetting is modeled—unless revisited or otherwise refreshed, a working memory item will cease to be available after a period of time and will have to be reacquired if needed again. In placing working memory items in a working memory store, the memory items are, in a sense, cut off from the context in which they were acquired. The absence of context for declarative memory items is a concern for which we sought a remedy. Declarative memory items are each acquired within a context and that context should be retained to accurately represent those memories.

Much of the work of the SO in processing the EEIs relies on declarative working memory. The SO's first action is to identify the target aircraft with virtually all of the SO's subsequent actions being related that aircraft—an element of the SO's working memory. The working memory item is revisited and hence refreshed repeatedly as the subsequent EEIs are prosecuted. Indeed, it would be hard to imagine a more simple and straightforward demand on working memory. The sixth EEI, surveillance of the airport, makes more complex demands; sensemaking (Weick, 1995) is required to interpret multiple observations occurring over a period of time and build a coherent story covering these observations. Pursuing the surveillance EEI includes a broader range of actions and makes more complex demands on working memory.

Actions taken in the form of the execution of procedures result in the accretion and possible forgetting of declarative memory items in working memory. Within this framework, *episodic* memory is the memory of the actions themselves—the model's memory for what the model did. In contrast to declarative working memory, human performance models seldom make allowance for episodic memory.

One of the more surprising aspects of the D-OMAR human performance models is that episodic memory is simply there waiting to be exploited in order to build better

representations of human-like behaviors. The instances of the procedures that execute are procedure objects, each based on an underlying SFL concept defining the procedure object. For a procedure to run, an instance of a procedure class is made, the procedure executes often over an extended period of time, and the procedure object persists after the procedure has completed execution. The retained procedure object is linked to the procedure by which it was initiated and the procedures that it subsequently invoked. As such, the persisting object associated with the completed procedure is the basis for an episodic memory item—the model's remembrance for what it has done.

Variations on how to exploit the persisting objects to create a realistic episodic memory for the D-OMAR models are actively being explored. With the episodic memory items in place, one of the first steps has been to construct accessors to reach back to the procedure objects to query them for the findings resulting from their execution. An accessor is triggered by a signal that identifies the procedure type and the particular finding sought; it provides a value in the form of a responding signal. For this to work, a procedure's findings are kept as slots on the procedure object making them readily accessible to the accessor. Declarative memory items that are the product of procedure execution are thus retained in the procedure objects in which they are acquired. The procedure object provides the context for the declarative memory items that are retained within the procedure object.

The procedure by which the SO model identifies the target aircraft, as illustrated in the upper left corner of Figure 4, retains the identification of the aircraft as a slot, Target Aircraft, with value N12345 on the procedure Determine Target Aircraft. Following the execution of the procedure, the procedure object is retained as the episodic memory of the procedure's execution and an accessor is created to support memory retrieval. The episodic memory object and the memory accessor are pictured on the right side of Figure 4. A procedure for subsequent EEI processing, as illustrated in the lower left corner of Figure 4, then makes use of the episodic memory accessor to retrieve the identity of the target aircraft as illustrated. The procedure, Monitor Target Aircraft, publishes a signal requesting the identity of the target aircraft. The memory accessor, having subscribed to the signal type, responds to the request by publishing a signal identifying the target aircraft that is then processed by the requesting procedure, Monitor Target Aircraft

In like manner, the procedure by which the SO counts the armed guard present similarly retains and makes available that count for later use in subsequent EEI processing. Further variations on the retrieval process are being explored. If the SO were to count the number of armed guards again, it is the new value that the SO would want to access from this later running procedure object rather than the originally determined count. On the other hand, when the SO is monitoring the actions related to two trucks servicing the target aircraft, a retrieval with respect to the trucks should return information with respect to both trucks. In the first case, a new declarative memory element within an episodic memory replaces a preexisting episodic memory and memory element; in the second case, the second memory item should coexist along side the first memory item each within its own episodic memory object.

For the present, an episodic memory retrieval return a particular requested slot value—a declarative memory item within the episodic memory for the procedure. Another variation being explored returns the procedure object—the episodic memory for the

actions taken. Extending the simple access to a declarative memory item within an episodic memory, the model can potentially further interpret the episodic memory for the previously executed actions. Having the SO model report on the key events of the mission would provide a serious challenge that would force the next steps in understanding and processing episodic memory in a human-like manner. The report would require an attempt at not just retaining the currently perfect episodic memory, but at developing a better understanding and an initial attempt at implementing a memory consolidation process. The consolidation of the remembrance of recent actions that extends and refines episodic memory is a key step within Glenberg's (1997) formulation of the retention and revision of what we know how to do as the central function of memory.

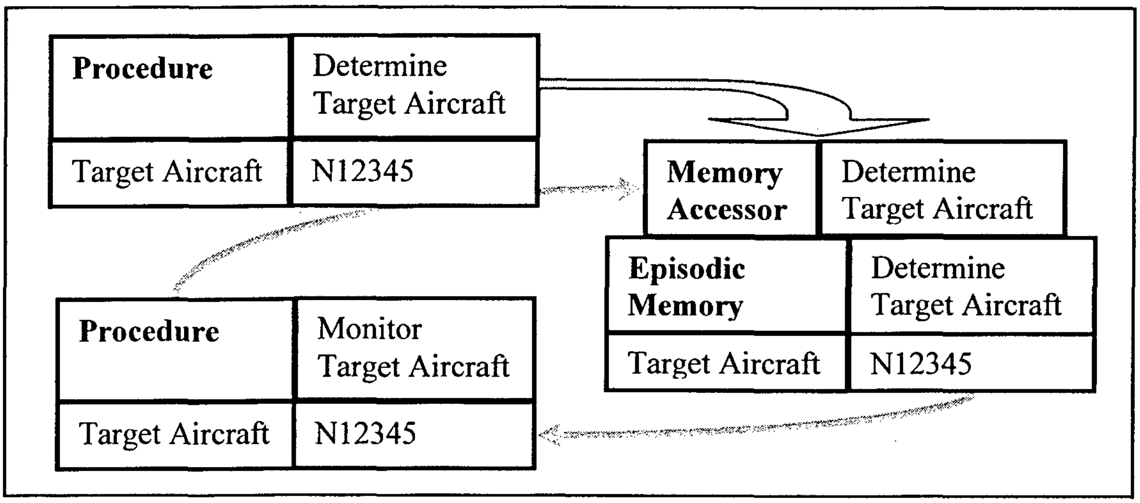

Figure 4 Memory for the target aircraft as an element of episodic memory

4.2.7 Robustness in the Sensor Operator Model

Robustness has been a long standing problem in artificial intelligence. Most often it takes the form of brittleness in system performance—a system may perform well when confronted with a given range of situational events, but exhibits an unacceptable behavior when encountering a closely related, but not exactly matching event. Much as robustness has been a problem in artificial intelligence systems, it is similarly a problem in the performance of human performance models (c.f., Deutsch, 2005a). In the early days of SIMNET, modeled tank platoons were not able to use the terrain effectively and often did not respond appropriately when negotiating a narrow bridge crossing under fire. While hopefully fixed by now, the narrow bridge crossing had not been adequately addressed when I last looked in on the problem several years after it surfaced. Individual robustness problems can be very difficult to address; providing robustness across even moderately challenging domains has proven elusive.

With untoward events such as these in mind, we were able to briefly examine the robustness issue with respect to the models that we had developed for the UAV scenario. In particular, we focused on the SO model as the most richly developed model in the

scenarios. As we began to look at model robustness, the SO model was successfully executing the six EEIs for the assigned scenario. Yet, robustness has readily been found to be wanting when there are small changes in scenarios events, or small changes in event timing as when model processes shift subtly in the timing of their execution.

The conflicting demands for using the TV camera for most of the observations and using the IR sensor to check for aircraft engine startup was an obvious point to probe in the SO's procedure execution. The model exhibited some robustness in that it coped nicely with missing the IR observation of the start up of the engines. In spite of having missed the engine start up, the SO model dutifully observed the movement of the guards as they boarded the aircraft, the aircraft as it began taxi operations, detected the aircraft as it moved onto runway five, and continued by observing the aircrafts departure from the airport. The model recovered nicely having missed engine start-up though IR sensor observations and recognized that it no longer needed to pursue that task.

Further probing did lead to a model robustness flaw in the sequence of TV-based observations of the guards entering the aircraft, the initial movement of the aircraft, and its movement to and departure from runway five. These TV-based observations were programmed to be executed and interpreted sequentially; first observe the movement of the guard toward the aircraft, and then each of the steps related to the aircraft's movements leading to its departure. Due the imposed sequentiality in observation and interpretation, when the model missed the movement of the guards toward the aircraft it did not move on to watch for the movement of the aircraft. By making small changes in the timing of the sequence of events leading to the start up of the engines, it was possible for the model to be occupied with an IR observation of engine start up as the guards entered the aircraft—the aircraft then moved toward the runway and took off, but the SO was left watching for the guards to enter the aircraft—the event that was missed due to the IR observations. The model that had worked well in its initial trials failed badly due to small variations in the timing of scenario events.

The model was revised so that when using the TV camera it concurrently observed the guards *and* the aircraft and watched the movements of each in a manner more like a person might actually effect the observations. Then having found the aircraft to be moving following a period of being preoccupied by IR-based observations, the model readily dropped the observations related to the movement of the guards moving toward the aircraft and concentrated on tracking the progress of the aircraft toward the runway.

The failure in robustness detected in the SO model related to issues of observation and interpretation. For the model to be more robust, it was important that the model observe more broadly and concurrently interpret the multiple aspects (within reasonable bounds) of the observed scene. When these extended capabilities were established in the model, the model's robustness improved. With these improved capabilities in place, we looked for another target in the model to which the same capabilities might be applied.

The Petkosek et al. (2005) scenario included a single cargo truck from which materials were being transferred to the target aircraft. Our SO model monitored this activity and duly reported on the initiation and completion of the loading of the aircraft and correctly anticipated the aircraft's departure. The process was linear in that each observation of the truck operation was followed by interpretation: having detected that the loading was

initiated, now check for the completion of the loading operation as depicted in Figure 5. At this point, operations with respect to the cargo truck were believed to be complete and the truck was no longer a subject for observation.

We then added another cargo truck to the scenario and had the driver moving cargo *from* the aircraft to the truck. As it turned out, the SO model readily observed the second truck but had to be refined to understand the movement of cargo to the truck rather than the other way. An implication of the flow of cargo to the truck was that the truck then continued to be an object of interest even after the aircraft departed. We now asked what would happen if an activity that appeared to have been completed was in fact continuing.

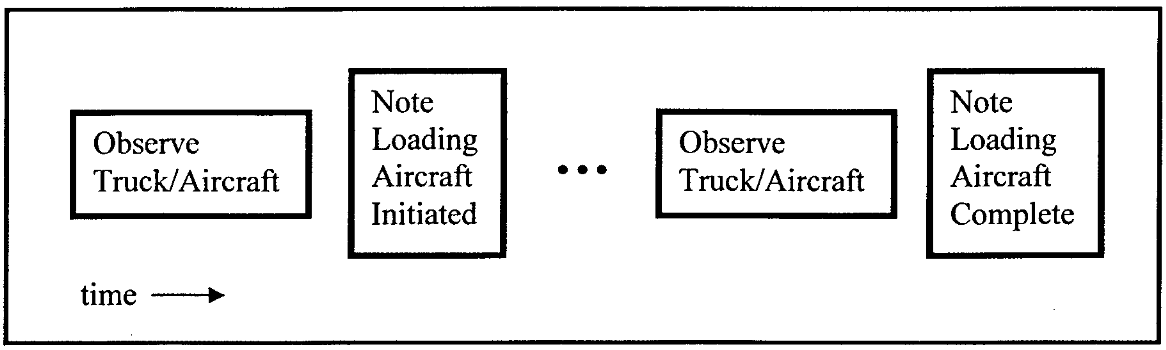

Figure 5 Monitoring cargo loading as a single linear process

The capabilities of concurrent, ongoing observation and interpretation, as outlined above, were also required here to properly interpret an evolving situation. With the appropriate changes in place, the SO model now returned periodically to the monitored the loading or unloading of each truck and concurrently developed and maintained an interpretation of the observed activities summarizing the observed events as depicted in Figure 6. The recurring observation and ongoing interpretation of the presented scene enabled the proper interpretation of the resumption of the loading of a truck whose loading had previously been interpreted to have been completed. Interpretation as an ongoing process enable the SO to adjust to more complex storylines in the activities now presented by the presence of multiple trucks. With the SO's continuing observation and interpretation of activities in the surveillance area the model was then able to correct its earlier finding that was in fact no longer valid. With the more rigorous approach to scene interpretation in place, the SO model was able to responded correctly to a broader range of activities related to the cargo trucks and the target aircraft.

Scene observation and interpretation were among the more challenging aspects of building the SO model. As the model was adapted to observe more broadly and engage in a wider range of interpretations, we were able to extend the number of players in the observed scene and thereby extend the number of variations in the way the actions could play out to further challenge the SO model. For the SO model, the need to move from the use of the TV sensor to the IR sensor and back to the TV sensor to execute the EEIs meant that the SO was more likely to miss a key observation based on one sensor or the other. A more comprehensive interpretation of the scene in a more human-like manner led to better recovery from missed observations of key events.

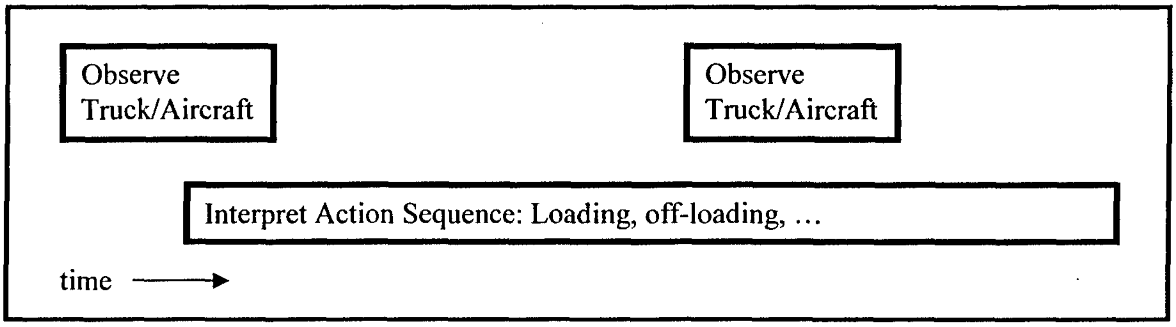

Figure 6 Observation and interpretation as concurrent ongoing processes

The interaction between observation and interpretation and model robustness is an interesting one that deserves more attention. Observing more of the scene and concurrent approaches to interpretation help with recovery from missed observations and allowed for the reinterpretation that avoided what would have been errors in interpretation. That we were readily able to move from one sequence in the scenario in which revised observation and interpretation led to improved robustness to another observation-interpretation sequence suggests that this is an area that should be further investigated. There are bounds on how much can be "seen" to be explored and with the greater complexity in the observed scene, interpretation will become more difficult. People are very good at sensemaking (Weick, 1995); this is an area that needs more attention in our modeling efforts if the models are to more faithfully represent people's capabilities.

4.3 The Multi-function Operator Model

The MFO plays a relatively minor role in the surveillance scenario as the person with whom the SO collaborates in pursuing the execution of the EEIs. In the use case scenario, it is the MFO to whom the SO reports his or her findings as the EEIs are executed. The AVO is the passive third party to these communications. The MFO also attends to the reports from the AVO about the progress of the UAV along the surveillance route.

The presence of the MFO in the scenario meant that there was then the requirement for three party in-person conversations, in contrast to previous modeling work in which in-person conversations were restricted to two parties—typically the flight deck conversations of a captain and a first officer. For the conversation model to work properly it was necessary to enable the speaker to direct his or her statements to a particular listener thereby cuing the listener that he or she was the appropriate person to rely. The real world is more complicated than this, but this addition to the conversation model proved adequate for the demands of the current scenarios. Three or more person conversations then became an integral part of the actions of many of the other scenario players, most notably on the part of the armed agent's leader as he/she directed the actions of the other armed agents and then conversed with the target aircraft's flight deck crew in preparation for departing the airport.

5 Potential UAV Test Bed Applications

In the design of new systems or seeking to remedy problems with existing systems, performance, workload, and error at the individual and team level now are being

addressed as basic cognitive engineering (Roth, Patterson, & Mumaw, 2002) concerns. Human-in-the-loop experiments have been a traditional method to determine these measures. Stud y metrics at the individual level measure performance, situation awareness, and workload; metrics at the system level address efficiency, capacity, and safety. Unfortunately, human-in-the-loop experiments have been quite expensive. Progress in human performance model research then made it possible to address cognitive engineering motivated solutions through model-in-the-loop studies. More recently, as modeling and simulation tools have improved and decreased in cost, improved flexibility in scenario development has led to a resurgence in the use of human-in-the-loop experiments.

While the current UAV test bed could readily be adapted to support human-in-the-loop experiments, for the present we will look at potential uses as a model-in-the-loop fast-time test bed. To illustrate potential applications, we will draw on recent research efforts for the NASA Ames Research Center in the commercial aviation domain. In each instance, the work involved the modeling of the operator workplaces—the aircraft flight deck and the air traffic controller workplace—and the procedures employed at these workplaces. A common theme in these research efforts was to draw on human-in-the-loop experiments for empirical performance data to support model development and validation, and then to extend the range of the investigations performed by employing the model-in-the-loop test bed.

The detailed process of constructing the models and their behaviors, in each case, led to new insights into potential improvements either in workplace design or the procedures employed. A series of human-in-the-loop experiments was conducted in which NASA examined the use of a Synthetic Vision System (SVS) that provided the Captain with a "clear day" out-the-window-like view under instrument meteorological conditions (Goodman, Hooey, Foyle, & Wilson, 2003). In the model-in-the-loop trials (Deutsch & Pew, 2004b) we were readily able to examine the use of an SVS by the Captain *and* the First Officer and particularly its impact in the face of a misalignment error in that view during final approach. In a similar manner, our work on the UAV test bed takes a small step in the direction of examining alternate workplace capabilities. The AVO model in our scenarios is using an FMC-MCP combination to manage the UAV's ingress to the target area rather than having to manually guide the UAV along the route. This higher level of control led to considerably reduced workload on the part of the AVO. It is representative of the type of equipment and procedural changes that can be readily explored using a model-in-the-loop test bed.

Good system design and operating procedures should also reduce the incidence of operator error and assist in error mitigation when human error does occur. Human performance models can play a role in helping us to better identify the potential sources of procedural errors. In a second NASA research effort, we modeled commercial aircraft surface operations (Deutsch & Pew, 2002) to further examine and explain a series of errors seen in NASA human-in-the-loop scenario trials (Foley, Andre, McCann, Wenzel, Begault, & Battiste, 1996; Hooey, Foyle, & Andre, 2000). Once again, empirical data derived from part-task experiments was used to support human performance model development. The improved understanding of the sources of human error derived from

the modeling effort was a first step in adapting systems and procedures to prevent error or, failing that, mitigate the effects of error.

There have been several studies that examined military UAV accidents (Manning et al., 2004; Williams, 2004). The UAV test bed has the potential act as base from which to support in-depth analyses of UAV accident scenarios that can lead to further insight into the sources for human error and then be used to pursue revised workplace design and procedural changes to prevent error and mitigate the effects of errors that do occur.

Several UAV accidents have occurred during launch and recovery mission phases (Williams, 2004). An innovative approach to this problem has been to establish special launch and recovery teams. The launch and recovery teams operate locally and the UAVs are then controlled from remote sites (Fulghum, 2005). Innovative tactics such as this are likely lead to improved operational performance.

One of the more difficult challenges has been to reduce the staffing required to execute UAV missions. The UAV test bed is a place where new ideas for reduced staffing can be explored. Piloting a UAV via FMC-MCP is a small step in reducing AVO workload. Multi-UAV control based, in part, on new approaches to UAV control in combination with some of the ideas being put forward for next generation commercial air traffic control might be explored in the test bed. In particular, the fly-out menus being explored for use by air traffic controllers are an innovation that might be adapted for AVO use in managing the routes of multiple UAVs.

Extending the launch-recover idea, a single ingress AVO might control several UAVs as they approach a target zone at which point specialists step in for more detailed control at the target area as necessary. Egress from the target area might be addressed in a similar manner. In the past, handoffs have been a factor in UAV accidents (Williams, 2004)—that the launch-recovery handoffs are being successfully employed suggests that handoffs are becoming less of a problem. The UAV test bed can be a good platform from which to examine new approaches to workplace design and the design of operating procedures for implementing new strategies and tactics.

6 References

Bargh, J. A. & Chartrand, T. L. (1999). The unbearable automaticity of being. *American Psychologist, 54,* 462-479.

Brachman, R. J. & Schmolze, J. G. (1985). An overview of the KL-ONE knowledge representation system. *Cognitive Science, 9,* 171-216.

Clark, A. (1997). *Being there: Putting brain, body, and world together again.* Cambridge, MA: MIT Press.

Deutsch, S. E. (2005a). *Robustness in human performance modeling: Theory building and architecture.* BBN Technical Report No. 8424. Cambridge, MA: BBN Technologies.

Deutsch, S. E. (2005b). Reconceptualizing expertise: Explaining an expert's error. *Proceedings of the 7th Naturalistic Decision Making Conference.* Amsterdam, NL.

Deutsch, S. E. & Adams, M. J. (1995). The operator model architecture and its psychological framework. *Proceedings of the 6th IFAC Symposium on Man-Machine Systems.* Cambridge, MA.

Deutsch, S. E., Adams, M. J., Abrett, G. A., Cramer, N. L. & Feehrer, C. E. (1993). *Operator model architecture (OMAR) software functional specification.* (AL/HR-TP-1993-0027). Wright-Patterson AFB, OH: Air Force Materiel Command.

Deutsch, S. E., Hudlicka, E., Adams, M. J. & Feehrer, C. E. (1993). *Computational cognitive models.* (AL/HR-TR-1993-0072). Wright-Patterson AFB, OH: Air Force Materiel Command.

Deutsch, S. E. & Pew, R. W. (2004a). *Modeling aircrew decision-making in D-OMAR.* BBN Technical Report No. 8387. Cambridge, MA: BBN Technologies.

Deutsch, S. E. & Pew, R. W. (2004b). Examining new flight deck technology using human performance modeling. *Proceedings of the Forty-eight Annual Meeting of the Human Factors Society,* New Orleans, LA.

Deutsch, S. E. & Pew, R. W. (2002). Modeling human error in a real-world teamwork environment. *Proceedings of the Twentieth-fourth Annual Meeting of the Cognitive Science Society.* Fairfax, VA

Foley, D. C., Andre, A. D., McCann, R. S., Wenzel, E., Begault, D. & Battiste, V. (1996). Taxiway navigation and situation awareness (T-NASA) system: Problem, design philosophy, and description of an integrated display suite for low visibility airport surface operations. *SAE Transactions: Journal of Aerospace, 105,* 1411-1418.

Fulghum, D. A. (2005). UAVs: The view from Nellis. *Aviation Week and Space Technology,* September 26, 2005, pp. 56-58.

Glenberg, A. M. (1997). What memory is for. *Behavioral and Brain Sciences, 20,* 1-55.

Gluck, K. A., Ball, J. T., Krusmark, M. A., Rodgers, S. M. & Purtee, M. D. (2003). A computational process model of basic aircraft maneuvering. In F. Detje, D. Doerner, & H. Schaub (Eds.), *Proceedings of the Fifth International Conference on Cognitive Modeling* (pp. 117-122). Bamberg, Germany: Universitats-Verlag Bamberg.

Goodman, A., Hooey, B. L., Foyle, D. C. & Wilson, J. R. (2003). Characterizing visual performance during approach and landing with and without a synthetic vision display: A part task study. In D. C. Foyle, A. Goodman, & B. L. Hooey (Eds.), *Proceedings of the 2003 Conference on Human Performance Modeling of Approach and Landing with Augmented Displays,* (NASA/CP-2003-212267), pp. 71-89. Moffett Field, CA: NASA.

Hooey, B., Foyle, D. & Andre, A. (2000). Integration of cockpit displays for surface operations: The final stage of a human-centered design approach. In *Proceedings of the AIAA/SAE World Aviation Congress*. Warrendale, PA: SAE International.

Logan, G. D. (1988a). Toward an instance theory of automatization. *Psychological Review, 95,* 492-527.

Logan, G. D. (1988b). Automaticity, resources, and memory: Theoretical controversies and practical implications. *Human Factors, 30,* 583-598.

Manning, S. D., Rash, C. E., LeDuc, P. A., Noback, R. K. & McKeon, J. (2004). *The role of human causal factors in U. S. Army unmanned aerial vehicle accidents.* USAARL Report No. 2004-11. Fort Rucker, AL: U. S. Army Aeromedical Research Laboratory.

McCarley, J. S. & Wickens, C. D. (2005). *Human Factors Implications of UAVs in the National Airspace*. University of Illinois Human Factors Division Technical Report AHFD-05-05/FAA-05-1. Urbana-Champaign, IL: University of Illinois.

Petkosek, M. A., Warfield, L. & Carretta, T. R. (2005). *Development of a human performance model of a UAV sensor operator: Lessons learned* (AFRL-HE-WP-TR-2005-0118). Springfield, VA: National Technical Information Service.

Roth, E. M., Patterson, E. S. & Mumaw, R. J. (2002). Cognitive engineering: issues in user-centered system design. In J. J. Marciniak, (Ed.), *Encyclopedia of software engineering* (2nd Edition). New York, NY: John Wiley and Sons.

Weick, K. E. (1995). *Sensemaking in organizations (Foundations for organizational science).* Thousand Oaks, CA: Sage Publications.

Williams, K. W. (2004). *A summary of unmanned aircraft accident/incident data: Human factors implications.* DOT/FAA/AM-04/24. Springfield, VA: The National Technical Information Service.